Release 1.0

87-8 10 August 1987

EXTENDING THE SYSTEM

Despite the Motorola 68000 lurking inside the Macintosh, Apple has always had to compete with IBM on the basis of software, not hardware. In the first Mac, Apple devoted the power of the 68000 to running systems software and the interface, not to running applications. This made each Mac easier to use, if slow. Moreover, Apple fiercely encouraged all the developers who wrote Mac software to follow similar conventions, which has resulted in further benefits in ease of use and in minimized configuration problems for third-party vendors, resellers and users. All these benefits have helped Apple and the Mac make headway against the entrenched IBM standard.

This accomplished, Apple is enhancing the performance of its machines while (more or less) adhering to the standards it has set up. But, from a different vantage, IBM is now catching up. With an 80286/386 inside and the promise of Presentation Manager outside, the PC/PS looks like a plausible machine for desktop publishing and all the things Macs are good for.

What can Apple do for an encore? The obvious answer is more software. The 68030, wonderful as it is, won't give Apple a big advantage -- certainly not a proprietary one. (The competition now includes Sun as well as IBM.)

One easy extension is MultiFinder, which gives the Mac limited multitasking capabilities with flaws that everyone will hasten to point out. In the end, the primary benefits to most end-users will be such mundane but meaningful advantages as data-sharing, print-spooling, and background file management and communications. It also provides the ability to switch quickly from task to task, even though other applications may be suspended. But IBM and Microsoft will shortly offer this capability too -- and without suspension of other applications.

MultiFinder has the advantage of running most existing applications, and the disadvantage that it doesn't do much to improve them individually. IBM/Microsoft's OS/2 will be far more disruptive; for that very reason it will enable and ultimately evoke a profusion of superior applications. Sometimes one needs a little commotion to create a market window enticing enough to lure people in. ⟹

THE TRANSCRIPTS ARE COMING!

INSIDE

EXTENDING THE SYSTEM 1
Apple's HyperCard.
PERSISTENT OBJECTS 5
The structure is the data.
Products & projects.
NITTY-GRITTY EXPERTS 17
AAAI debriefing.
AI landfill.
Tell me a story...
RELEASE 1.1: THE SQUIGGLE 20
RELEASE 0.5 21
dSTANDARDS.
Tandy is dandy.
Softwords from IBM.
Menage a trois.
TroisCom buys Bridge.
PHONE NUMBERS/CALENDAR 24/25

In the functions it adds, MultiFinder is closer to a control program such as DESQview or Microsoft's own Windows (ex compatibility problems), than to OS/2. It allows multiple existing programs to share memory but not always processing power, and it provides good facilities for inter-process communications and background operation for developers who make the effort to use them. The achievement of this sort of multi-tasking is relatively easy, because Mac applications by definition rely more fully on the operating system than most DOS applications do, so there's little of the "misbehavior" that makes it so difficult to provide concurrency for DOS applications on the 286. (With its support of multiple virtual machines, the 386 makes that feat a little controllable.)

Link construction kit

Far more exciting and controversial is HyperCard, the unfortunate new name of Wild Card (adopted because other companies also claim Wild Card). HyperCard, a powerful facility that deserves to be on every Mac, draws uncomfortable attention to Apple's long-time conflict between its need to compete with IBM and its need to support third-party software developers. (IBM suffers from the same conflict, but has the market power mostly to ignore it.)

What *is* HyperCard? Well, it has some flavor of hypertext (hence the name) but in limited enough form that true hypertexters will sneer, while non-adherents will still be frightened by the name. HyperCard consists of the ability to build pointers or links between "objects" (or to build hypertext) enhanced with a rudimentary file manager. The basic HyperCard object is the card. The cards are records which are displayed one to a screen and can be sorted and searched by the file manager. Other objects include stacks, or sets of cards; fields on the cards; and buttons, which activate the links. All of these may have associated procedures -- application modules in traditional parlance, or "behaviors" in objectspeak.

For example, an activated button might flash a dialogue box such as "What model is your car?" and then display the appropriate picture for you through a link to the appropriate stack (and a card retrieved by the file manager). These application modules can be edited, or "specialized," so that with very little effort a user can create customized applications and links from application to application. Although multiple buttons can appear on a single card, and one card can hold pointers to another card ad infinitum, only one card can be seen at a time, which makes the structure difficult to follow.

Although Apple's publicity appropriately describes the power of HyperCard -- and the power it gives to the user -- it's a little overenthusiastic in its notion of hypertext as free association. It's a way to *record* free association so that it can be retraced, but one user may expect to move from Federal Express to UPS, while another may want more information on overnight rates. Certainly, the buttons that represent the links can be labeled, say, "details on services," "competitors," "earnings report," "overdue bills." But if the developer of the software doesn't free-associate the way you do, tough luck. To quote Mac designer Jef Raskin (now president of Information Appliance, **Release 1.0**, 9 July): "One man's link is another man's sausage."

Missing links

HyperCard lacks the ability to embed objects within text -- something Owl's text-based Guide offers. HyperCard limits you to jumping from screen to

screen (although text will scroll within each field displayed on a card), while Guide has no file management capabilities. Long-run, Stackware[tm] vendors should add spreadsheet and text objects, as well as a relational dbms, so that one could have both explicitly built links and massive joins among large amounts of data, based simply on matches between items in similar fields in different tables. With this, one could avoid the tedium of building all the links by hand -- and end up approaching an object-oriented dbms.

Despite different positioning and capabilities, Framework on the PC provides some vision of what HyperCard could become. HyperCard is more limited in many ways, without spreadsheet or word-processing objects, although richer in others, with its extensive linking and much richer graphics (what did you expect?). Framework allows you to see much more of the structure you have created, with lists of objects (frames), while the only way to inspect objects within HyperCard is to move from one to another along a tortuous path that's hard to hold in one's mind.

The value of hypertext in general and of HyperCard in particular (vs., say, an outliner) is its nonlinearity. Yet the user's experience of HyperCard is almost totally linear: He can go forward or back, and branch at any point, but he has no way of surveying the scenery that surrounds him. It is like trying to navigate with a map that shows you only one crossroads at a time. Perhaps one of the first pieces of Stackware[tm] will be a little program that collects objects and displays them in a little list -- or better yet, WebWorks, a graphical map that shows their interconnections. (With a half-gigabyte stack (HyperCard's limit), or many stacks, that would be some map!)

Weak links

Also limited so far -- and dependent on the good offices of third-party vendors -- is the ability to go easily from the HyperCard environment into other applications, which right now is restricted to launching another application and opening a specified file. When you leave the application, you return automatically to the card you left from. Richer interconnection and sharing of HyperCard's capabilities will happen only when other application vendors build appropriate connections. That is when the system will become truly an extension of the operating system rather than a minimalist stand-alone application. (Along with WebWorks, we're hoping to see LookLink, akin to pop-up notes, which lets you automatically link into a HyperCard stack and its associated procedures, and go on from there.)

HyperCard is not an application but a facility with a couple of applications thrown in, in much the same way that a dbms is a facility. (IBM's bundling of a dbms into OS/2 Extended is another expression of that notion.) Because HyperCard is hard to explain, it is best off sold as an extension of the operating system, so that people don't need to make a decision about it: It's just there. Meanwhile, software developers will be able to develop Stackware in the knowledge that they can make use of the HyperCard facilities on each user's Mac. Much rides on developers' confidence that HyperCard will continue to be bundled for several years, and that Apple will successfully induce existing Mac users to upgrade to it. (We hope that it will ultimately move into the Mac ROM, although Apple has no such plans. Apple might also consider providing runtime or even full versions free for developers to include with their applications. The $49 charge for upgrades to HyperCard presumably covers distribution costs that would thus be covered by the application vendor.)

Overall, as the typical user becomes more sophisticated, the demarcation between application and tool shifts. More users are prepared to build their environments to suit, to use software to express themselves. What seemed impossible a few years ago -- users creating their own customized data bases, screen layouts, dialogues and application routines -- will soon seem as normal as the notion of an individual driving from Long Island to Long Beach -- no longer a pursuit of pioneers, but an ordinary event unworthy of comment. The infrastructure provides roads and maps and gas stations; most people own cars and learn to drive as a normal part of growing up.[1]

Of course, what some people consider flexibility, others consider lack of standards or disarray, and they want their applications pre-built for them. For those who are doing the work, there are approaches such as SoftLogic Solutions' @Liberty and Brubaker's Baler 3.0, which compile 1-2-3 models into fixed applications. This is the embodiment of the notion that Lotus 1-2-3 is an environment -- so let's make those models into static applications that cannot be embellished (i.e. abused) by their users.

Too-tight links?

With its creation of Claris (and its indirect creation of Acius, the company founded by former Apple evangelist Guy Kawasaki to sell 4th Dimension in response to pressure from Ashton-Tate and others), Apple has attempted to clarify its understanding of the distinction between applications and systems software. It could easily enough have given HyperCard to Claris -- and probably would have but for its agreement with creator and Apple Fellow Bill Atkinson to bundle the product. Vendors who consider it an application -- because they're selling a file manager or a hypertext builder themselves -- will indeed find their livelihood threatened. But, with all due concern for people's feelings (and pocketbooks), we don't find this so upsetting. Buggy whip makers adjusted; so did makers of disk managers and graphics software when those became part of the system; so did makers of wp and paint/draw programs, who were encouraged to do a better job than they might have otherwise to compete with the Apple offerings; and so will the file manager and hypertext folks, such as Owl International. The smart ones will get busy making Stackware, building and improving on the facilities in HyperCard.

But the controversy over Claris is likely to heat up further as the company becomes active in acquisitions. Says Bill Gates, chairman of a well-known Macintosh software house (which sells both systems and applications software): "[Through Claris,] they now have executives whose compensation depends on screwing us." *Intending* to spin something off has few of the advantages of ownership and all the disadvantages of lame-duckitude. Apple would like to hand over the burden quickly and no doubt applauds Claris ceo Bill Campbell's sentiments: "As soon as we look like a real company with our own arms and legs, we'll get disengaged." But that's not likely to happen until 1988, if only for logistical reasons. Meanwhile, Apple has turned its investment in Forethought over to Microsoft, and has pledged not to invest in any other application vendors until Claris is cleared out.

---

[1] *We note that the Adonis Theatre on Broadway is showing an all-male movie, "Hard Disk Drive Therapy: Silicon Valley Meets West Hollywood," conclusive evidence of the computer industry's integration into the mainstream -- and beyond!*

PERSISTENT OBJECTS

Meet my daughter the object base administrator.
-- a proud mother

Why are you going to be hearing about object-oriented data base management systems? Not because relational data base management systems can't do the jobs they were designed for, but because there are new jobs to do. For example, Computervision wants to let its customers think about walls, pipes and stresses, not about points and coordinates. A software vendor wants all his documentation to have the same look and to suit each user's configuration. Interactive Development Environments, a CASE vendor, wants to let systems designers know immediately which modules and test routines are affected when they change a data definition somewhere. And they all want to let their users share not just their data but the *structure* of their data.

With object-oriented programming, you can have many of these benefits, but until now object-oriented programming has been like traditional programming before the advent of data base management systems: The structure of the data was all mixed in with the applications. In those old days, data had a fixed but simple structure; it lived in flat files, and the applications could define further structure, expressed (unintelligibly) in procedural code intricately tied to the data. With the advent of data base management systems, the data became independent of the applications, and it could now be represented in richly structured (but inflexible and hard to maneuver) network or Codasyl data bases. That enabled you to change data independently of the application, but you still couldn't do much about the structure.

Relational data bases put some inherent structure into the data base (in the form of matching fields among tables), but they leave the interpretation of the structure in the applications, which can select the data they need to form appropriate *logical views*, generated at runtime by relational joins based on matches between items in separate tables. That is, you can create different logical structures out of the same set of physical flat files.

Now we're moving to a third iteration, putting much more structure back into the data base, but leaving it flexible and independent of the application and the system environment. We will call it an object base; it consists of the object management system (also called an object-oriented dbms) plus the instances of the objects it describes. The object management system manages the schema -- a rich logical representation of the data's structure along with the knowledge of how to store and find the data. (Just as the schema of a relational data base consists of a table listing all its tables, so does the schema of an object management system consist of a hierarchy encompassing all the object classes in the system.) With such an object-oriented approach, you can store both the data and the associated knowledge -- procedures, validation criteria, relationships -- in the object base independently of the application, which needs to know very little (as opposed to a relational dbms application, which needs to know the names of the relevant tables even though it may not need to know their locations). A procedure in an object base could even include the appropriate query to get particular data from a traditional dbms, thus providing links to existing data bases. (For a full discussion of the benefits of object-oriented programming, see **Release 1.0**, 24 March.)

(continued on page 7) ⟹

OBJECTS ON PARADE

The *schema* (structure) of an object management system is hierarchical: An *object* is akin to a table in a relational dbms, except that objects have parent-child relationships. There are *classes* and *subclasses* of objects; a class's *attributes* are *inherited* by its subclasses. *Instances* of objects are like individual records; and attributes (or *slots*) are analogous to fields, but rather than just data they can also hold references to other objects or to *methods*, or *procedures*. (The procedure may be a way to respond to a particular *message* (such as instructions on how to display itself, which would vary for different graphical objects) or simply a way to determine a value (multiply the values of two fields, or run a little program). Specific instances of the objects, with particular values for the attributes, exist in the object base managed by the schema (except for in the AI model, where types, instances, values and methods are mingled together in a primordial soup). For example, here's an instance of "sales employee":

sales employee: Alice Haynes
age: a number [or perhaps a birthdate with a procedure for finding age]
employer: The Objective Organization, Inc. [an instance of the "company" object, to which it is linked]
department: sales
salary: base + (0.1*sales) [a procedure from the sales employee object]
sales: [a procedure to search a corporate data base]
customers: WareWorks, Spinoff Software,...[also objects themselves]
instrument: flute

There are three specific classes of relationships (so to speak):

is a -- an instance to an object
Alice is an employee. Alice is a musician.

is a kind of -- an object (class) to its (super)class
A sales employee is a kind of employee.

is a part of -- an instance to another instance
The sales department is a part of a company.

an undefined relationship -- an instance to an instance
Alice's employer has a value of The Objective Organization, Inc.

In this example, Alice has derived all but her last slot from the sales employee object. The sales employee object inherited the first five slots shown from the employee object, a superclass of sales employee. Alice derived the last slot from the object musician. Some object bases can handle such a single *instance* of multiple objects, or *multiple data types*, but few can support *objects* with multiple parents, or *multiple inheritance*. That is, you couldn't create "musical sales employee" as a subclass of both employee and musician. In the last two relationships shown, an object may have a slot for a reference to another object, but the relationship itself is between the instances.

==========> *(continued from page 5)*

However, all this is easier said than done. Only one of the object management systems described here is actually on the market: G-Base, from a French company, has 18 installations in France, and one sale in the U.S. Few commercial users expect to do much more than prototypes until 1989.

Object management systems in use

To use object-oriented terminology: In this article we first attempt to establish a framework (or schema) for classifying the various kinds of object management systems, noting the attributes (values and methods) by which they can be identified and assessed. Then we offer a number of instances, with specific data, starting on page 13.

Object management systems aren't going to be much help near-term in posting general ledger transactions or handling reservations for an airline. The tasks in those systems are too performance-intensive to work on an object base and (by default) simple enough not to need one. However, the task of modeling such a system, or of designing the airplane the customers will fly in or planning their trips, would suit an object management system just fine. For example, says Bill Harrelson, director of data management at Computervision, his company will soon offer a restricted version of an object management system within one of its application packages. Longer-term, Computervision is investigating the use of an object management system as a substrate integrating a number of separate design applications. Object management systems shine at structure-intensive tasks where the goal is to hide complexity, while in transaction-based systems the ultimate goal is speed, which for now at least means a sacrifice in richness and flexibility. Objects let the user deal with data as abstractions -- without worrying how the record is going to be found or how the correct carrier is determined. (You can, of course, inspect the object to find out these details -- and change them -- but you don't have to.)

Appropriate applications include design, geographical or other spatial data bases, software engineering, and the like. Try following which stock-market winner knows someone who knows someone who sits on a board with someone married to someone who works for a company that has just been taken over; the SEC might be interested (or the NSA, for different purposes). Or try rotating all the separate elements that constitute a 10-speed bicycle in a design data base, for instance, or selecting all the cities within 100 miles of Chernobyl. Unless you've designed a traditional data base with this particular query in mind (with a field for "distance from Chernobyl"), that will be a tough one. In an object-oriented data base, by contrast, you could easily add a method for determining distances, taking the location in question as an argument, that could painlessly provide the result for the user.

It's not that you couldn't use the same procedure within an application, but that would limit its availability and generality. One of the values of object management systems is to keep the data independent of the applications, but not to rely on the applications to represent relationships that are true *across* applications.[2] (Among other things, this fosters reuse of code.)

---

[2] *Sybase, for example, espouses putting some data-related procedures (which it calls triggers) into the data base, but avoids the other trappings of object-orientation, such as hierarchical inheritance and structures.*

In a commercial vein, consider a travel agency: It might want the flexibility to deal with customers individually, yet to combine their reservations in certain cases to earn discounts. Each individual's trip might or might not include plane, hotel and car reservations, plus car-and-driver, special requests and the like for certain individuals. These complexities could be nicely handled in an object base. Each individual may have preferences, but would also have to observe some restrictions depending on his company and employee class. The object base might also include procedures for linking to the various reservations systems, so that complicated bookings could still be done automatically: It would be easy to express such notions as second-choice hotels or alternatives, "If he can get a flight out of Palm Springs on Sunday, then book him at the Bel-Air Sunday night; otherwise..." However, object-oriented programming per se is still procedural, and a rule-based paradigm would be best for this example.

How this all would happen is invisible to the user, who doesn't need to worry about how the records are stored. True, the system builder has to understand the application intimately, but even he will find it far easier to extend or modify such a system than one with a traditional architecture.

Appropriate architectures

As usual, proponents see the object management system as suitable for everything, while less biased observers consider it a valuable tool for modeling and classifying applications, or for simple applications needing large amounts of complex data, such as our who-knows-whom data base. The question is, Is the value of what you're building in the structure, or in the data it contains? If you care most about the data, use a regular dbms (preferably relational, these days); if you care about the model, use a spreadsheet (for mathematical relationships) or an object management system for symbolic relationships.[3]

If you care deeply about both the data and the structure, you can combine the two approaches, with varying degrees of success and integration. The data that represents specific instances of objects may or may not be stored within the object management system. Some people take dbmses and attempt to extend them with object-oriented ideas, while others take object-oriented languages and try to render the objects persistent. The dbms group treats the object-oriented model as an extension of a traditional data base, to which it is linked with pointers (cf. HP's Iris, Berkeley's Postgres).[4]

---

[3] *A spreadsheet is a particular, limited kind of object management system, rich in relationships albeit lacking in hierarchy or heterogeneous types. All the objects are cells, containing labels, numbers, formulas or macros. A spreadsheet can be extended (cf. Symphony and all the 1-2-3 add-ins) but the results are usually somewhat clumsy. Framework, by contrast, is an object-oriented dbms of considerable richness, with objects such as spreadsheets, graphs, documents, text elements and associated stylesheets, although its hierarchical structure is restricted in that each object can have only one parent.*

[4] *Strict relational data bases handle only a limited range of types, so that, for instance, a field can be validated as being an integer within a certain range. With the addition of triggers (as in Sybase) or full-fledged abstract data types as in an object base, data can be validated by procedures as complex as the designer can make them. In an rdbms, such validation would be accomplished by a separate application, such as one run once a week to find deadbeat accounts.*

At the other end of the spectrum, more AI-oriented object management systems tend to be implemented with classes and instances stored together -- that is, making no distinction between types and instances of types. This makes sense for applications such as modeling, but not for production systems with large numbers of instances of each type. The AI-oriented object management systems also tend to support multiple inheritance -- which allows for richer structures but makes schema modification more difficult and perilous.

With trade-offs here and there, all classes of object management systems provide far richer structuring, as well as the hope of some efficiency; inheritance means that not all the relationships need be built by hand, and that they can be compiled for efficiency. At either end of the spectrum, a dbms could consist entirely of relationships among objects without inheritance, or pure hypertext; or all strict hierarchies with no undefined relationships and no multiple inheritance -- the kind of object management system that's easiest to build and manipulate.

In fact, systems at either end of the spectrum are the only kinds currently on sale. Combining the advantages of both is diffcult. (Apple's new HyperCard (page 1) combines some features of both, but its data management capabilities are limited.) But in all cases, much cross-application structure -- whether it is expressed procedurally as in traditional applications or in rules as in an expert system -- can be kept within the object base for use by all applications and users.

Implementation issues

To date, a number of people have discovered the virtues of object-oriented programming, but they have been restricted by the lack of a general-purpose object management system. So far the only way to combine object-oriented programming with large numbers of objects is to use large amounts of memory and virtual memory. This explains in part why object management systems are relatively unused: They are easy to use, but they consume system resources heavily. Object-oriented systems to date are generally small and limited in number, restricted mostly to prototypes or special-purpose implementations.

What's the difference between using an object base and simply storing all the objects in mass memory? Without a data base there is no easy way to get any particular object when it isn't in memory, reachable through pointers also in memory. Following pointers becomes a tedious process when each link requires a new disk access (which will be a problem with CD ROM-based hypertext systems). People can't easily share their objects (classes) either in real-time on a network or for use with different applications.

Object management systems offer the ability to index the data and compile the relationships among them for easy and efficient access. And, like other dbmses, object management systems hold the promise of automatic resolution of conflicts among users, to say nothing of the mundane issues of data integrity, error handling, optimized storage and the like. These issues seem mundane, but the traditional dbms world has been wrestling with them for years. Facilities to address them matter a lot when they're missing, as they still are in most cases. And they're essential if object-oriented programming is to gain sway in the commercial world.

(continued on page 11) ⟹

OBJECTS ON DISPLAY

Institution /Product	"camp"	application language	platform	status
Berkeley Postgres	db	C	UNIX wkstn	research project
Computer Corp. Probe	DB	C	UNIX wkstn	DARPA project
Graphael G-Base	AI	LISP	LISP wkstn	on sale
Hewlett-Packard Iris	DB	C, LISP OSQL	HP wkstn/ PC	lab project
MCC Orion	AI	LISP C	LISP/UNIX wkstn	research project
Ontologic Vbase	OOP	C plus extensions	UNIX wkstn	beta
Servio Logic GemStone	OOP,OA	Smalltalk C	VAX/PC/ wkstn	beta
Symbolics Statice	AI	LISP	Symbolics	in development

The demand for object management systems is just starting to evoke a supply, with a growing number of companies and other institutions in various stages of the cycle from project to product. They come from three camps: data base, AI, and object-oriented languages. They divide most sharply into those addressing the small but eager AI/ Smalltalk market: Graphael, Symbolics and MCC; and those hoping to sell to the larger but less receptive mainstream market: Ontologic, Berkeley/Relational Technology, and Hewlett-Packard. Despite its use of Smalltalk, Servio Logic is also addressing the commercial marketplace. Other outfits involved include IBM, Bell Labs, Texas Instruments, Brown University, the Information Sciences Institute of the University of Southern California, the University of Colorado, and the University of Florida. (For fuller descriptions of those in the table, see page 13.)

We have specifically avoided listing capabilities here, because most of these systems are changing daily. Moreover, it is not appropriate to compare the promises of prototypes with the testable if clumsy implementations of deliverable products.

(For the curious, Santa Clara University will hold two seminar courses for non-students on object management systems, including specific comparisons of Vbase, Iris, and GemStone, in December. See page 29 in the calendar.)

⟹ *(continued from page 9)*

Objects in storage

Like any data base management system, an object management system serves to provide a logical view of its data to users and to applications, while storing the data physically in the most efficient way. This uses three fundamental techniques: indexing, clustering, and compiling. Indexing simply provides a way of finding the data, and is accomplished in the usual way, with B-trees and hash tables and simple lists. As in data bases, indexing uses the system's structures and also values of individual instances.

Clustering is technically more interesting, and implemented in a way more specific to the object-oriented schema. Clustering stores logically related objects close to each other, both to provide quick access and to improve the hit rate when objects are cached. A user's access will be improved to the extent that the items he needs are present in physical memory (cached). And even when the items are on disk (either locally or across a network), access will be faster if related items are stored together. Determining what's "related," of course, is the crux of the matter, especially when the relationships get complex. Optionally, the system builder can provide some guidance, based on his knowledge of the system's likely use, or the object management system can rely on built-in optimization techniques. The process is similar to clustering in a relational data base, but based on different criteria (potential joins, generally).

Compiling renders an object base even more efficient, but like any such fixing process, it exacts a trade-off cost in flexibility. Compiling is early binding: I.e., rather than inherit their attributes from superclasses or instantiate relationships across a network of pointers, some objects and instances are stored "pre-bound." The compiler is sensitive to data locations and can organize things upfront for efficient execution, so that the application does not have to follow a series of pointers at runtime but can instead get directly to the data it needs. That is, you might find Alice's location, a function of her department's location, stored with the rest of her data, rather than have to find her department and then *its* location. This is fine until you want to change some relationships or modify your classes -- and is why early users give Vbase (a heavy user of compilation) good marks on access time but poor ones on schema modification time.

Server vs. local architectures

Servio Logic, which comes from an office automation perspective and seats the user at a PC, differs from most of its competitors in offering its object management system in a server configuration. (Servio also lets the user operate in "local" mode as a terminal to the VAX, using the OPAL language.) This is a fine distinction, generally visible only to builder-users rather than end-users. To the extent that the distinction matters, the server model probably sacrifices a little speed of access for the ability to offer the user a more natural environment. Additionally, given the processing overhead, the server approach is better for handling small numbers of large objects (such as documents) with applications that don't appear specifically "object-oriented," while the local approach is better for handling large numbers of small objects and making heavy use of the object's individual procedures for accomplishing much of the work.

Objects in transaction

Transactions, known as split-second modifications to a record in the commercial dp world, frequently take a different form in the object world. A transaction might be a design modification that takes several hours as Alice tinkers with the intricacies of constructing a ten-speed gear on a bicycle. Meanhwile, Juan should be free to change the bicycle's color from red to green. But what if Tom wants to add a footbrake in a way that will interfere with the gears?

As in the traditional dbms world, there are two basic approaches: pessimistic and optimistic. Optimistic says. "Assume everything will be all right," and then lets Tom know when he tries to store the footbrake that Juan has invalidated what he's been working on. The pessimistic approach tells Juan and Tom that the bicycle is in use, and locks it. A modified version warns them that Alice has checked the bicycle out, but lets them go ahead anyway. A fine-grained system may let Juan change the bicycle's color while keeping Tom from messing with objects attached to the rear axle. All this depends on the facilities of the object management system combined with the restrictions designated by the object administrator.

A third approach, possible in a full-featured object management system, is version management, whereby several different versions of the same instance of an object can be stored, including versions of parts of it. Conflicts can be resolved, or the desired set of parts and versions (Tom's footbrake and Juan's color) can be selected, by a designated person or a clever program. This lets the object base manage inconsistencies and configurations without blowing up, an important capability for most design applications. Creating and then resolving inconsistencies is a fundamental part of design -- not a process that should be aborted by an error message.

Objects in proportion

Contrary to what you might expect, large objects are not necessarily good. Consider this analogy: Large subdirectories hold a lot of information, but make it difficult for a user to find anything in particular. There is more information and more utility in a hierarchy of subdirectories, each holding a smaller number of files. In the same way, a large object leaves it up to the user of the object to define most of its structure, whereas a schema of smaller objects incorporates more built-in information (and takes more memory). In a similar but less rigorously analogous way, a long, featureless document can be tougher to read than one leavened with chapters, pictures, boxes, subheads, and the like. (See page 6, and also pages 10, 19, and 24.) The visible structure itself provides part of the meaning.

Objective advice

Once you've got your object management system, most of the value lies in the specific objects and procedures you create, and for performance, the indexes you build. That is, building the object base requires programming, optimization, careful design, and all those other manifestations of human intelligence and perspiration that produce good software of any character. Object management systems don't offer a free lunch; they just make it easier to design the menu.

G-base from Graphael

Graphael, owned by engineering services vendor SGN of St. Quentin, France, has just opened a U.S. subsidiary in Waltham, MA, after a half-year of unsatisfactory efforts to sell its product G-Base through distributors for whom it was just one more product on a list. The U.S. division made its first sale in June to a LISP machine vendor (not a broad range of possibilities!), and has been busy responding to a barrage of follow-up interest generated at the AAAI show last month. G-Base is written in LISP, runs on LISP machines and LISP-equipped Sun workstations. G-base's graphics are extremely rich; it has browsers and tree-editors so that you can see the data structures you are building and using. As yet, the product is still single-user, single-inheritance, and lacks version management. It costs $25,000 per copy, including three days of training and one day of on-site support.

Graphael U.S. president Joe Hancock, who learned the "AI" business as a regional sales manager for Texas Instruments, is aiming strictly at the "AI" market for the moment, including the installed base of KEE (IntelliCorp, roughly 1600 copies), ART (Inference, 700), and KnowledgeCraft (Carnegie Group, 300) that lack a way to store their knowledge bases. In France, Graphael has sold 18 copies since 1985, into a target market of 70 LISP machines. Its customers include Renault, for a manufacturing application, and the French Navy, which is using G-base to track ships and, in conjunction with ART, to maintain submarines.

Hancock hopes eventually to build links to traditional dbmses, but for now the market looks appealing as is. Graphael is also talking to Gold Hill about a PC-based version running on top of GC LISP and linked to GoldWorks.

GemStone from Servio Logic Development

Servio Logic Development, in Beaverton, OR, is an affiliate of Servio Group in San Francisco. It was funded five years ago by an Indonesian businessman with U.S. operations who needed a way to keep track of them. However, the company is mysteriously reticent about him and its customers. (Local observers watch the company with interest.) About the product GemStone itself, however, the Servio people are refreshingly open, to the point of listing its deficiencies in a set of technical papers delivered at conferences and published in academic journals.

For all that, Servio is one of the three companies listed here that has a real product, even though it is currently only on beta test at nine universities and national labs. Nor will it discuss pricing, although we expect the price to be in five figures, like any VAX dbms, with adjustments for size of host and number of users.

GemStone's developers come from the Smalltalk world, but they have designed their product to handle commercial applications such as office automation and inventory management, rather than "workstation" applications such as CAD/CAM and software engineering. For that reason it works (optionally) with a PC workstation, and manages multi-user access, rather than such nice-

ties as multiple inheritance and version management, which would make the system slower, bigger and tougher to use. Senior product manager Mike Connell (ex-TI and IBM) sees the market shifting slowly from academic/government research to corporate research to commercial use over the next couple of years. (On this scale, Ontologic is already further ahead, with a number of corporate customers.)

The server portion runs on a VAX with VMS, with the user portion on a Smalltalk workstation or on a PC running Digitalk's Smalltalk or Windows and C. The object schema is built in Servio's language, OPAL. Applications are assumed to be written in Smalltalk or C, with embedded OPAL statements and queries. GemStone takes the optimistic approach to concurrency control, working on the assumption that most users will need access to information (such as checking inventory status) but only a few will be involved in changing it. In the future, Servio plans to offer optional pessimistic control, whereby a user can check out an object and keep it from others' hands.

GemStone is server-based, providing easy access to a number of users, but optimized for a lesser level of granularity, than, say, Vbase. Given its PC support, the examples used in the documentation and the company's whole approach, it's likely Servio Logic will be most successful in non-object-oriented applications, such as office automation, electronic publishing, and data management where the data is complex but the applications possibly quite simple, as in our who-knows-whom data base.

The company hopes to finish beta testing by the end of this year.

Vbase from Ontologic

Ontologic of Billerica, MA, is the new incarnation of Mosaic Technologies, a workstation start-up that foundered a few years ago. Bob Berkowitz, formerly vp manufacturing at Prime, was brought in to save or close down the company in 1985. The company's computer no longer made sense, he determined, but in the ashes he found the basis of an object management system, with which he won new funding ($7 million) from Alan Patricof, Kleiner Perkins and others. The company is now valued at approximately $10 million. Seventeen of 22 signed customers currently have beta copies of the software, and Berkowitz hopes to attract another 30 before revenue shipments start next January. The system currently runs on Sun micros and will be ported to the VAX under VMS by September.

Berkowitz is painfully familiar with the problems of tracking and classifying huge numbers of objects from his days managing inventories at Prime. From Prime he also brings the notion that the very best products are not niche-market, but general platforms suitable for a wide variety of functions. Accordingly, he plans to price Vbase at about $10,000 on a single-user workstation (to start) and in the $40-80,000 range on hosts such as VAXen and IBM mainframes.

Vbase itself is being shipped in single-user, beta form, with the multi-user version to be ready for production in January 1988. It still lacks the graphical browsers and other enhancements of its competitors, although those are scheduled for the September 1988 release, along with version management and multiple inheritance. For now, the focus is on functionality, as repre-

sented in a customer list of names that includes United Technologies and Tektronix. While Computervision is not a customer, that company expects to be working with Vbase on a factory-floor automation project at the National Bureau of Standards' Automated Manufacturing Research Facility.

While GemStone and Hewlett-Packard's Iris (below) have been designed with user-friendly interfaces and query languages aimed at making them accessible to end-users, Vbase is positioned as a tool for software houses who will incorporate it into their products, or internal corporate builder-users who will make it the underpinning of internal systems. The company is also working on a way to hide Vbase so that it could fit under existing applications, replacing a traditional dbms. In the short run that means easy adoption of the new paradigm. In the long run, the flexibility and power of Vbase should encourage the development of new applications that can then use those hidden data structures.

Vbase is written in C and easily portable across a variety of hardware platforms from its current Sun base to mainframes and PCs eventually, vp of development Bob Strong promises. Although all information is preliminary, indications are that Ontologic is making significant progress in increasing the raw speed of Vbase. Accesses are reasonable, says one user, but defining new data structures (which requres recompiling) is still quite slow.

Altogether, Vbase does the best job of integrating object-oriented paradigms with the traditional dbms world. Although we list it in the object-oriented camp, the product and its positioning place it firmly in the commercial environment -- although convincing customers of that may still be tough.

Statice from Symbolics

As the leading vendor of high-end symbolic development environments, Symbolics keenly feels the lack of a standard way to store persistent objects, and has been working for some time to build an object management system to enhance its Genera operating system. This product, Statice, at this point is still a development project led by technical director Dan Weinreb. Very rich in functionality (and like most Symbolics product cleverly named), Statice gives the user access to a rich LISP-oriented environment with strong facilities for virtual memory management, multiple inheritance and version management, although it does not yet have compilation or clustering for efficient storage that it will need for release. Statice may be released sometime next year, but the company is making no promises.

Iris from Hewlett-Packard

At present, Iris is a project of Hewlett-Packard Labs, meaning that there are no formal plans to sell it as a product. However, its developers, led by Dan Fishman, manager of the Database Technology department, have taken punctilious note of such issues as interfaces to existing environments. Iris sits on top of the same relational storage manager that underlies HP's commercial relational dbmses -- HP-SQL and the newest implementation of the popular Image. Written in Pascal and C, Iris has its own SQL-like query language, OSQL (Object SQL), and runs on an HP 68000 UNIX workstation, with optional access from a PC or other front-end. Already multi-user because of its multi-user foundation, Iris has a limited form of multiple inheritance.

Orion from MCC

Orion is a research project of MCC's Systems Technology Lab which underlies many other efforts at MCC, such as Proteus (an expert system shell), natural language, and human interface design (with graphical objects). It is also under consideration for use in a software engineering system. However, Orion is not part of Doug Lenat's CYC project (**Release 1.0**, 14 March 1986), which is a knowledge base designed to be so rich and associative that it will remain stored in virtual memory. Like the other AI-oriented projects listed here, Orion does support multiple inheritance, but it does not yet manage multiple-user conflicts. In other words, it's a lot more flexible than the commercial products, but it provides no safeguards against misuse.

The eight sponsors of the MCC group working on Orion include DEC, Control Data, Bellcore, Kodak, Harris and NCR. They have had access to a copy of Orion since last April, and several are using it internally, says project manager Won Kim, for tasks such as CAD, expert systems, and software documentation management.

Postgres from Berkeley

It takes a rich model of the world to explain the significance of Postgres, which is more than just another academic project. Postgres is the follow-on to Ingres, which began as a project at Berkeley and ended up as the core of Relational Technology Inc.'s commercial relational dbms offering. Although the graduate students have changed, the Ingres project leaders, Michael Stonebraker and Lawrence Rowe, are now working on Postgres, and are founders of and consultants to Relational. Postgres is in the public domain, but Relational Technology, located nearby in Alameda, is likely to have an edge in bringing some form of Postgres to commercial life. The company talks about the project to customers, and expects to incorporate some of its notions into Ingres over the next couple of years. These notions include abstract data types, support for storage of maps and text, and some expert systems constructs.

Ingres was started in 1974 and launched by Relational Technology in 1981; Postgres was started in 1983. However, in this case a company stands ready to commercialize the research, and perhaps the market will be more receptive than it was in 1981 to an unknown vendor of an untested product -- a relational dbms.

Probe from Computer Corporation of America

Probe, a project of Computer Corporation of America's Advanced Information Technology division, is the successor to CCA's Daplex; both are DARPA-funded projects ($1.5 million over 30 months for Probe) to explore the design and uses of object management systems. Funding is forthcoming for a third stage of the effort, and CCA computer scientist Jack Orenstein hopes it may eventually reach a commercial stage. So far, Probe has been used in prototype mapping and air-traffic control applications. It runs on a UNIX workstation and uses an internally built relational dbms for data storage.

NITTY-GRITTY EXPERTS: AAAI DEBRIEFING

The American Association for Artificial Intelligence held its annual conference in July, and only 5026 paying people came, down from 5400 the year before. The program, recovering from a brief flirtation with commercialism in 1986, focused heavily on academic discoveries and concerns. "Mainstream" press attendance was also down. Many of the "AI" companies present are hurting, showing losses, turnover, and other turmoil. Inference has laid off several salespeople, in response to a weakening it sees in the high-end product market. It will be spending relatively more resources on consulting than on product sales. IntelliCorp will announce a substantial loss this week and has reduced staff by 10 percent to focus on conventional platforms and vertical markets.

Is this the long-awaited AI winter? Not quite. In fact, you could read AAAI as a sign of health: Many of the AI companies have moved on to commercialism, leaving behind the academics and the researchers to carry on the research. While it may not be the "AI" companies that benefit, the notions they have propounded are reaching beyond the research community into dp rooms and executive suites.

Accountants are using expert systems to minimize taxes, and Wall Street houses are using them to maximize revenues. Aion is exhibiting at the Info show; Syntelligence spoke at NCC, at an Alliance of American Insurers conference on underwriting, and at other insurance industry trade shows. MSA and Cullinet are incorporating expert systems modules within their traditional applications. Artificial intelligence is one track amidst such mundane topics as decision support and software productivity at conferences sponsored by Digital Consulting and Decision Support Technology for audiences of information systems and dp managers. These are the same people who wouldn't dream of attending a conference on new developments in data base architecture -- yet they use data bases and order their creation and modification every day.

NITTY-GRITTY EXPERTS: IN SEARCH OF LANDFILL

One of the hot topics at AAAI among would-be commercial vendors is the notion of a "bridge" to conventional dp environments. Offerings include IntelliCorp's KEEconnection (shipping shortly), as well as movements onto traditional hardware platforms and into traditional languages. But all these approaches still require developers to do hand-coding to link the expert system to data stored in the traditional fashion (as do the object management systems described below). To be sure, the situation is no different between any two different traditional environments, but customers have long been complaining about that too -- and they're hopeful that the widespread adoption of SQL will ease that problem at least as far as data is concerned.

But bridges in general are inadequate and slow. By contrast, the embedded-rule systems such as those offered by Aion and DMS (**Release 1.0**, 8 July), the enriched data structures soon to be offered by Ontologic and Servio Logic, and possibly Teknowledge's Copernicus (shipping this winter) and Artificial Intelligence Corp.'s KBMS seem likely to take a substantial share of the market. They're closer to what we're holding out for -- a landfill!

NITTY-GRITTY EXPERTS: TELL ME A STORY

"This article is about all of the above" doesn't make much sense all by itself, but it's the sentence you'd get if you filled in the blank below correctly.

This article is about _____.
(a) expert systems;
(b) electronic publishing;
(c) automatic text generation;
(d) fill-in-the-blanks software;
(e) all of the above.

This article is actually about how to avoid problems like the one we just demonstrated. There are several approaches to generating text of varying degrees of complexity, all of which can be used in conjunction. They start with fill-in-the-blanks, the more general form of mail-merge, where a document is merged with specified data from a file, and form letters are generated. The corresponding angle is form fill-in, where a user enters data into a form with fixed fields (see also **Release 1.0**, 2 December 1986, "A new paradigm..."). Mail-merge can get fairly complex, with conditional fields and the like, but the overall structure of the document produced is usually fairly fixed: The selected text modules are organized within the document.

Next is text assembly, where chunks of text are assembled into a document. This approach tends to be less fine-grained than the preceding one, and is typically used in legal offices: "Use paragraphs 1, 3, 8, 15 and 20, and add a clause after paragraph 8 specifying that the cats are to live in the house for 10 years or until it is sold, whichever comes first."

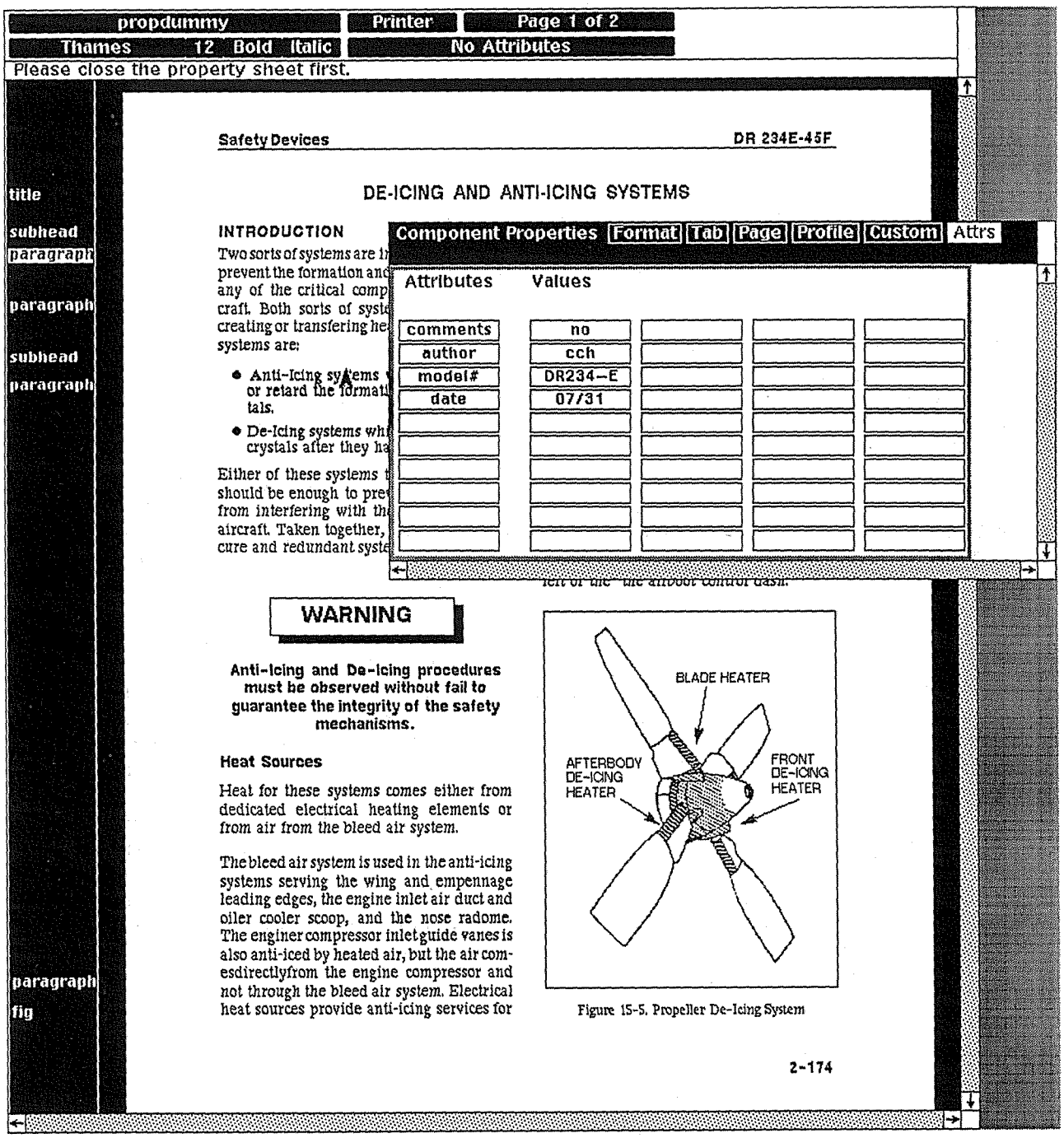
This can be done by hand, by a clever paralegal, for example, or the process can be automated, as in Version 4 of Interleaf's publishing system, scheduled for shipment this winter (and announced last spring). Interleaf helps reduce the complexity by structuring it. As each module of text is entered (or altered), the person responsible fills in a table, specifying various parameters. In the lawyer's case, for example, they might consist of states where the statute explained by the module applies, the marital status of people to which something applies, etc. In the Interleaf screen shot shown on page 19, the paragraph referenced applies only to icing systems for model "DR234-E." A corresponding table is filled out at runtime, so that a document with modules tailored to fit the criteria specified can be produced.

Another approach is The Living Office's Document Modeler, which helps people design a highly interactive, conditional Q&A format rather than a form to elicit the same sort of information. (See **Release 1.0**, 24 September 1986.)

In all these cases, the structure of the resulting document is generally straightforward, following the fields checked off or the answers given to the queries. In other words, the text is inserted into a template, which determines its overall structure.

Expert systems meet automated text

But suppose you want to generate text to handle a more complex set of criteria. You want something that will create the appropriate structure, and then something to ensure that the text within it flows smoothly and reads fluently. Specifically, suppose you want to generate a letter of advice to an investment client, with a complicated financial situation, two children



Here is an Interleaf screen shot, with a dialogue box where the author ("cch") has already entered some attributes of the paragraph indicated by the pointer and the highlighted reference in the left margin. (In an object-oriented system, a single reference might be able to incorporate the subheading and the associated diagram as well, but in this system each component must be identified and classified separately.)

in need of college educations, an ex-wife and a current wife, and a second home in Westport. At some point the "select one from column A and two from column B" model breaks down. And if he's wealthy, he's not going to look kindly on a letter that discusses his "child(s)" and advises him to

mortgage your home(s) to make the payments on your ex-wife's alimo(s).

Fill-in-the-blanks just doesn't work for complex situations, and the language it produces frequently looks terrible.

Higher-level structure, lower-level consistency

One company, Applied Expert Systems of Cambridge, MA, has spent several years addressing these problems (**Release 1.0**, 2 October 1985 and 25 January 1987), both with expert systems built especially to handle the intricacies of personal financial planning and with a language-generating tool to make the resulting advice sound human-generated. With the expert system assembling the appropriate text modules, the language generator ensures that the resulting sentences read fluently. It knows, for example, about plurals and possessives, pronouns and antecedents (as marked in the text by the developer), and it can express such rules as "when the value of 'fortune' is greater than $50 million, delete section 'Public Transportation'." In fact, such complex rules are generally best expressed within an expert system, while the text generator does the fine-tuning -- matching cases and tenses, for example.

Apex considers the knowledge in its expert system, developed with a number of financial experts, highly proprietary, but it has decided to make its text-generation capabilities available to all -- for a fee, of course. (Those text capabilities, incidentally, were developed not by a bunch of AI hackers but by a team of technical writers led by software engineer Terry Smith.) To reach beyond its own market of financial planners and the institutions that employ them, Apex has turned to Gold Hill Computers, which will start reselling the package under license starting this winter. Apex Computed Text^{tm} (trademarked and patent-pending) uses Gold Hill's GCLISP on PC ATs and other 286/386 machines and cooperates nicely with Gold Hill's GoldWorks. (Interfaces to other expert systems will require a little work by users.) Computed Text will cost $2000 to $3000, with runtime pricing still unclear. (Even a runtime system, of course, is in a tool for creating models to create text rather than an application itself.)

Apex Computed Text, of course, is not magic. In fact, it's a rather complex language that requires a sophisticated user to build a document generator. But the results do indeed (a) read as well as regular text, or better, if the regular text is not properly edited; (b) produce consistent responses; (c) save tremendous effort once the template is built; and (d) all of the above.

RELEASE 1.1: ABOUT THE SQUIGGLE...

On page 4 of our July 9 issue, the printer erased the squiggle we had so carefully drawn in. (Natural intelligence, huh?) This is how it should have looked -- a cross between a 2, 3, 7 and Z.

dSTANDARDS

A group of dBASE/Dbase third-party vendors, which together sell about 14,000 compilers and interpreters per month (or perhaps a third of Ashton-Tate's volume), has gotten together to establish a standard for the Dbase language. Normally, we'd expect such a group to have as much success as did Compaq, AST Research, Chips & Technologies and the other companies rumored to be working on a response to the PS/2. In that case, however, there was both a genuine new contender, IBM's PS/2 and Micro Channel; and a plausible response, the existing standard, that required no group effort. In the case of the Dbase language, however, there's something of a vacuum -- Ashton-Tate offers no compiler. The group's goal is not so much to create a standard as to discover what it is by carefully reverse-engineering dBASE III Plus to determine the precise specifications -- which aren't very clear in the documentation, says group leader Martin Rinehart, chairman of WallSoft. (See also **Release 1.0**, 16 January.)

TANDY IS DANDY

It takes a smart company to build computers for not-so-smart people. (See **Release 1.0**, 8 July.) The most impressive thing about Tandy's new line is the care it has lavished on making its computers accessible not just with pricing, but with interfaces. The operating system (DOS) and Tandy's Personal DeskMate friendly shell are now in ROM on Tandy's 80286-based 1000 TX, so that the user need do absolutely nothing beyond pushing a button to get into a friendly menu that will guide him through his software. Other models have the same software, but not all in ROM.

Why are local-area networks most frequently found in larger companies? Not because smaller single-digit-pc companies don't want or need them, but because they can't figure out how to install and use them. By addressing those problems, Tandy has made a significant contribution to the cause of networking that is likely to go rewarded with increased sales. Tandy is now offering TandyLink, a low-cost AppleShare-compatible but three-times-faster network. Better yet, it has enhanced its Professional DeskMate DOS shell (now partly in ROM) with DeskLink -- call it a network shell: A suite of useful network applications (shared calendars and phone directories as well as the usual E-mail and peripheral-sharing). The higher-end systems get real Ethernet from 3Com, another beneficiary of Tandy's good sense.

Notes Tandy's Ed Juge, "Our proprietary technology goes into our low end." After all, any*one* can buy an 80386.

SOFTWORDS FROM IBM

As Apple spins off its software application unit to mollify third-party developers, IBM is taking the opposite route, attempting to make its application efforts more coherent and more visible. IBM's software revenues have already increased from 5 percent in 1982 to 8 percent in 1985 and 11 percent in 1986. But this ratio still leaves IBM as a hardware company in a world where the growth, the value-added, the account control -- that is, the profits -- reside in applications software. The new Applications Systems Division bespeaks IBM's understanding that it must become a significant

player in the software business long-term both to generate revenues and to maintain account control. The message is intended not only for third-party vendors and customers, but for IBM's own troops. "IBM does not create divisions easily," notes Bob Berland, ASD's new vp of vendor and development operations. In fact, Entry Systems Division, transformed from an independent business unit in 1983 in recognition of the PC's success and importance to IBM's overall direction, is the most recent other example.

This move is only the most visible in a series of efforts to raise the company's presence in the software market, including the announcement last March of Systems Application Architecture, a standard/specification set which will be much easier to define and enforce from a single source (**Release 1.0**, 24 March). IBM has also launched its SolutionPacs, which make application software easier to buy and install, whether it be a support-included Expert Systems SolutionPac or a printer-included offering for secretaries. And over the past couple of years, the company has entered into a profusion of alliances with software vendors ranging from Hogan Systems (banking) to Manac (law practices).

Contrary to some popular opinion, ASD is not simply a vehicle for for further alliances and resale agreements. "There's an avalanche of people who read the papers and call up and say, 'Here's a great idea, IBM. Wouldn't you like it?'" says Berland.

But in fact the goal of ASD is not to corner the market but to improve the company's relations with third parties by providing better support, technical information and training; there's no way IBM could ever hope to handle the same market share in software, with its heavy requirements for support, that it has in hardware. ASD is in part an attempt to leverage IBM's systems software with third-party development efforts and support. Yet to the extent that IBM *does* resell products, it is likely to concentrate more on workstation, server and mainframe applications, where it has an edge in providing value-added, than stand-alone PC applications, where all it can offer is a name.

Third parties regard these moves with some suspicion, all the while hoping to be one of the anointed chosen to co-market with IBM. In fact, the market is still large enough and hungry enough that IBM's activities will do more to sell hardware (and software) than to take share from other software vendors.

MÉNAGE À TROIS

Borland's acquisition of Ansa surprised and amused the world, which knows of few more different people than elegant, polished Ben Rosen and loose, flamboyant Philippe Kahn. Ron Posner may be a lot closer to Philippe -- certainly they both have an inordinate fondness for overseas travel and a propensity to make deals -- which may in fact mean that there's less room for Posner in the new company than there is for Rosen, who will complement Kahn nicely.

In fact, the merger is not as preposterous as it might seem. Ansa has been in the market for a corporate partner for some time, in an effort to gain the critical mass that has eluded it so far. We're sure the investors were

disappointed that they couldn't be the *acquiring* party... Meanwhile, Borland saw an opportunity to gain the classy image that had eluded it, as well as a corporate salesforce, enabling it to speak directly to a market where its products are well-represented. Will Borland start to raise prices now? It has just almost tripled the price of Reflex for the Mac (nee Interlace, **Release 1.0**, 32 December 1985) to $280, a level that will enable dealers and perhaps Borland to support it effectively in competition with 4th Dimension, among others.

For those surprised at the mingling of the Kahn and Rosen styles, remember that Mitch Kapor's primary credentials were as a TM teacher and a disk jockey, with "a little programming," when he first met Ben Rosen. Says Rosen: "There aren't many left like that now that Mitch is gone. Philippe is one of the few technocrats [besides Bill Gates] still running a major software company."

DOWN THE AISLE WITH BRIDGE

The recently announced merger of 3Com and Bridge makes far more sense than either of 3Com's previous abortive attempts to merge with Convergent (still single) and Centram (now part of Sun). In fact, the rationale is so straightforward -- each company had half a product line, with 3Com on the low, pc end, and Bridge on the high, SNA, wide-area network end -- that we'll address the more interesting issue of channel conflict.

3Com has heretofore vigorously espoused the dealer channel, and highlighted its commitment in contrast to Novell's willingness to sell direct. Novell, of course, already has the product line (gained through a series of acquisitions and strategic alliances) that 3Com hopes to gain by merging with Bridge. But 3Com must now deal with dealers who will feel threatened by the presence of Bridge's direct sales force. Of course, Bridge was already selling 3Com products under a resale agreement with 3Com, which gave 3Com far less control than ownership will. The smart dealers will also note that they can now hope to sell to companies who want an upgrade path from the same vendor -- and that they will probably pick up a lot of aftermarket business from Bridge customers who will now be sold on the virtues of buying from 3Com. It cuts both ways...and bigger (approaching $200 million) does mean stronger in this market, all other things being equal.

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers the pc, software, CASE and connectivity markets, and the transformation of artificial intelligence into a commercial technology. Editor & publisher: Esther Dyson; associate publisher: Sylvia Franklin; circulation manager: Hyacinth Frederick; copy chief & consulting editor: William M. Kutik; office manager: Lynne Miller; proofreader: Mia Dyson. *Copyright 1987, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we will gladly arrange for reprints or bulk purchases.* *Subscriptions cost $395 per year, $475 overseas; multiple subscription rates on request.*

PHONE NUMBERS

K. C. Branscomb, **Aion**, (415) 328-9595
Terry Smith, Bill Woods, **Apex**, (617) 492-7322
Bud Colligan, Charles Oppenheimer, **Apple**, (408) 996-1010
Bob Goldman, **Artificial Intelligence Corp.**, (617) 890-8400
Philippe Kahn, **Borland International**, (408) 438-8400
Judy Estrin, **Bridge Communications**, (415) 969-4400
Bill Campbell, **Claris**, (408) 973-4583
Jack Orenstein, **Computer Corporation of America**, (617) 492-8860
Bill Harrelson, **Computervision**, (617) 275-1800
Carl Wolf, Stan Curtis, Gene Wang, **Gold Hill**, (617) 492-2071
Joe Hancock, Yannick LeNoan, **Graphael U.S.**, (617) 890-7055
Dan Fishman, **Hewlett-Packard**, (415) 857-4906
Howard Hunger, **IBM Applications Systems Division**, (203) 783-4410
Alex Jacobson, **Inference Corp.**, (213) 417-7997
Tony Wasserman, **Interactive Development Environments**, (415) 543-0900
David Weinberger, David Boucher, **Interleaf**, (617) 577-9800
Bob Balzer, **Information Sciences Institute**, (213) 822-1511
Dave Winer, **Living Videotext**, (415) 964-6300
Won Kim, Gene Lowenthal, **MCC**, (512) 338-3463, 338-3439
Bill Gates, **Microsoft**, (206) 882-8080
Bob Strong, Bob Berkowitz, **Ontologic**, (617) 667-2383
Aaron Zornes, **Relational Technology Inc.**, (415) 769-1400
Mohammad Ketabchi, **Santa Clara U**, (408) 554-2731
Mike Connell, Monty Williams, **Servio Logic**, (503) 644-4242
Rick Cattell, **Sun Microsystems**, (415) 354-4679
Bob Epstein, **Sybase**, (415) 548-4500
Dan Weinreb, **Symbolics**, (617) 621-7500
Bill Krause, **3Com**, (408) 562-6400
Ed Juge, **Tandy**, (817) 390-3549
Earl Ecklund, **Tektronix**, (503) 627-5244
Jean-Pierre LeJaq, **United Technologies**, (203) 727-7670
Martin Rinehart, **WallSoft Systems**, (212) 406-7026

COMING SOON...

- *Connectivity: Promises, promises.*
- *Agenda managers -- a new category.*
- *Text management.*
- *Announcements from Lotus, Microsoft.*
- *Nitty-gritty experts.*
- *The Forum transcripts.*
- *And much more...*

RELEASE 1.0 CALENDAR

July 14 **Thank you Bill!**

August 17-18 **Software futures: An executive forum** - Boston. With James Martin, Carma McClure, George Schussel, Eric Firdman, Larry DeBoever, Paul Hessinger. Sponsored by Digital Consulting Inc. Call (617) 470-3880.

August 23-28 **IJCAI** - Milan, Italy. Tenth International Joint Conference on AI. Call John McDermott, Carnegie-Mellon, (412) 268-2599.

August 26-28 **Interactive Videodisc in education and training** - Washington, DC. Sponsored by Society for Applied Learning Technology. Call Raymond Fox at (800) 457-6812 or (703) 347-0055.

August 31-September 2 **ISO development seminar** - Monterey, CA. From vapornet to how to build something real, in three days of detailed exposition. With experts on all the layers of X.X. Sponsored by Advanced Computing Environments. Call Dan Lynch at (408) 996-2042.

September 1-3 **PC Expo** - New York City. Topics, exhibits for micro managers. Sponsored by PC Expo. Speakers including Gerry "Focus" Cohen; Irene Greif, Lotus; Gordon Eubanks, Symantec; Mike Maples, IBM. Call Steve Gross, (800) 922-0324 or (201) 569-8542.

September 1-4 **Interact '87** - Stuttgart, W. Germany. Second IFIP conference on human-computer interaction. Contact: H. Bullinger, Interact '87, Fraunhofer-Institut fur Arbeitswirtschaft und Organisation, Holzgartenstr. 17, D-7000 Stuttgart-1, West Germany.

September 3 **Novell Inc. at the analysts'** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Judy Zatz, (212) 344-8450.

September 8-16 **DECWORLD** - Boston. The annual extravaganza, sponsored by DEC itself for all its friends. Contact: Bob Kucharavy, (617) 264-1018.

September 9-12 **The desktop publishing conference** - Santa Clara, CA. Sponsored by Seybold Seminars. Last year, Steve Jobs declared HP "brain-damaged;" this year, who knows? With John Sculley, Paul Brainerd, John Warnock. Contact: Lena Lillsunde, (213) 457-5850.

September 12-13 **Softeach** - New York. Training/pitching for a profusion of products, classroom style. Sponsored by Softsel. Call Laura Keezen at (800) 325-9189 or (314) 827-1724.

September 13-16 **Breakaway '87** - Orlando, FL. Sponsored by ABCD: Microcomputer Industry Association. Contact: Terri Rojas, (312) 693-2223.

September 14 **Scientific Atlanta at the analysts'** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Judy Zatz, (212) 344-8450.

September 14-15 **The realities of manufacturing automation** - San Diego. Sponsored by Dataquest. Contact: Marsha Newman, (408) 971-9000.

September 14-16 **DataStorage87** - Santa Clara, CA. Sponsored by DISK/TREND and Freeman Associates. *The* conference in the field, with Larry Boucher of Adaptec, Ryal Poppa of Storage Technology, and others. Contact, Cartlidge & Associates, (408) 554-6644.

September 14-15 **Expert systems in financial institutions** - New York City. Sponsored by Institute for International Research. With David Shpilberg, Coopers & Lybrand, and representatives from Syntelligence, Apex, Palladian; Booz, Allen; Athena Group; and many user organizations. Contact: Linda Martin, (213) 278-9380.

September 14-18 **Electronic printer and publishing conference** - Miami. By CAP International. Call Jean O'Toole, (617) 837-1341.

September 15 **CACI International at the analysts'** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Judy Zatz, (212) 344-8450.

September 15-17 **ICCC-ISDN '87** - Dallas. Sponsored by International Council for Computer Communication. Understand ISDN, once and for all. With Thomas Bolger, ceo of Bell Atlantic. Call Jane Farthing, (703) 974-5435.

September 16 **MSA at the analysts'** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Judy Zatz, (212) 344-8450.

September 17 **Computer Sciences at the analysts'** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Judy Zatz, (212) 344-8450.

September 18-20 **AppleFest** - San Francisco. Back after three years of terminated non-residence. With Del Yocam, Steve Wozniak, Alan Kay, Martin Engel. Sponsored by Cambridge Marketing. Contact: Vidar Jorgensen, Andy Calkins, (617) 860-7100.

September 19-20 **Softeach** - Atlanta. Training/pitching for a profusion of products, classroom style. Sponsored by Softsel. Call Laura Keezen at (800) 325-9189 or (314) 827-1724.

September 20-22 **Agenda 88** - Carlsbad, CA. Stewart Alsop's first annual conference, managed by Dottie Hall and Tricia McGinnis. With Bill Gates, John Sculley, Vern Raburn, Heidi Roizen, Tim Mott, Victor Alhadeff, Scott McNealy. Call Elizabeth Readerman, (415) 363-8080.

September 20-24 **Design management conference** - Martha's Vineyard, MA. Sponsored by the Design Management Institute, with special attention to product design and packaging -- but not just computers. Contact: Betsy Speer, (617) 236-1315.

September 21 **Fall membership meeting of the Massachusetts Software Council** - Newton, MA. With a program on fourth-generation languages. Contact: Joyce Plotkin, (617) 497-5716.

September 21-23 **Computer-aided software engineering symposium** - Boston. With CASE stars Rich Carpenter, Ken Orr and Ed Yourdon, with two days of vendor presentations. Conducted by Digital Consulting. Contact: Daniel Horgan, (617) 470-3870.

September 21-23 **Conference on software maintenance** - Austin, TX. Sponsored by several professional societies. Contact: Roger Martin, National Bureau of Standards, (301) 921-3545.

September 24 **Western Digital at the analysts'** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Judy Zatz, (212) 344-8450.

September 23-25 **Systems Forum** - San Diego. Sponsored by PC Tech Journal. With Bob Epstein (Sybase), Rich Bader (Intel), Alan Ashton (WP), Gene Wang (Gold Hill), Steve Ballmer (Microsoft), Mike Swavely (Compaq), Philippe Kahn, Craig Burton, Bob Metcalfe and others. Call Marti Cunha, (301) 740-8300.

September 27-30 **ADAPSO management conference** - Colorado Springs, CO. With David Shpilberg (Coopers), Bob Flast (American Express) among others. Contact: Sheila Wakefield, (703) 522-5055.

September 29 **CEO roundtables of the Massachusetts Software Council** - Newton, MA. Get together to talk about common problems. Contact: Joyce Plotkin, (617) 497-5716.

September 29-October 2 **Info Show** - New York City. Sponsored by Cahners Exposition Group. Contact: Frank Fazio, (203) 964-0000.

October 1-2 **Campus computing '88** - San Francisco. "Marketing conference for computer executives." Sponsored by PCW Communications. Contact: Shirley Gines, (415) 546-7722.

October 4-8 **Computer Services and Communications & Information Systems Seminars** - Baltimore. Sponsored by Alex. Brown. The standard in Wall Street software events, combined with communications and information systems this year. Call Chris Mortensen, (301) 727-1700.

October 4-8 **OOPSLA '87** - Orlando, FL. The second annual conference on object-oriented programming, sponsored by ACM and chaired by Adele Goldberg (ParcPlace) and Chet Wisinski (PPI). With Kurt Schmucker, Mike Nastos, others. Contact: Jerry Archibald, (914) 789-7695.

October 5-7 **Strategic issues in managing information technology** - Boston. With Pat Winston, Esther Dyson, T. Capers Jones, Michael Scott Morton, David Ness, and Larry Meador. Sponsored by Decision Support Technologies. Contact: Donna Kacin, (617) 354-6400.

October 10-11 **Softeach** - Chicago. Training/pitching for a profusion of products, classroom style. Sponsored by Softsel. Call Laura Keezen at (800) 325-9189 or (314) 827-1724.

October 11-14 **SPA fall conference** - Washington, DC. The Software Publishers Association gathers for a roster of increasingly business-oriented sessions. Contact: Ken Wasch, (202) 452-1600.

October 13-15 **PC Expo** - Chicago. Topics and exhibits for micro managers. Sponsored by PC Expo. Call: Steve Gross, (201) 224-5835.

October 17-18 **Softeach** - Toronto. Training/pitching for a profusion of products, classroom style. Sponsored by Softsel. Call Laura Keezen at (800) 325-9189 or (314) 827-1724.

October 17-18 **Computer Professionals for Social Responsibility annual meeting & banquet** - Cambridge, MA. With keynote by economist Lester Thurow: "Does military spending hurt America's economic performance?" Contact: Gary Chapman, (415) 322-3778.

October 19-22 **CASExpo** - Washington, DC. Sponsored by Arthur Young & Co. Contact: Jim Duchaine, (703) 845-1657.

October 20 **Mergers & acquisitions seminar** - New York City. Conducted by Broadview Associates; sponsored by Adapso. Contact: Kelly Bailey, (703) 522-5055.

October 20-22 **Networld** - Dallas. Sponsored by Novell. Contact: Annie Zdinak, (800) 237-0316 or (201) 569-6916.

October 21 **Venture capital/Initial public offerings seminar** - New York City. Conducted by Accel Partners; sponsored by Adapso. With the voices of experience from Harry Reinstein (Aion) and Craig Hill (Cortex). Contact: Kelly Bailey, (703) 522-5055.

October 24-25 **Softeach** - London. Training/pitching for a profusion of products, classroom style. Sponsored by Softsel. Call Laura Keezen at (800) 325-9189 or (314) 827-1724.

October 25-29 **ACM-IEEE Joint Computer Conference** - Dallas. "Exploring the knowledge-based society." Contact: Cindy Cegelski or Kerry Kirschbraun (exhibits) (214) 746-3547, or Stephen Szygenda, (512) 471-1653 (conference).

October 26-28 **CEPS FALL** - Boston. Corporate electronic publishing systems. Sponsored by Cahners Exposition Group. Contact: Nancy Walsh, (800) 255-7798 or (203) 964-8287.

October 27 **Mergers & acquisitions seminar** - San Francisco. Conducted by Broadview Associates; sponsored by Adapso. Contact: Kelly Bailey, (703) 522-5055.

October 27-29 **Computer-aided software engineering symposium** - San Francisco. See September 21-23. Contact: Daniel Horgan, (617) 470-3870.

October 27-29 **UNIX expo** - New York City. Managed by National Expositions. Contact: Don Berey, (212) 868-2727, or (212) 391-9111.

October 27-30 **Educom** - Los Angeles. "Toward a global information culture: Education, libraries, and technology." Sponsored by Educom, with speeches by John Sculley, Simon Ramo, and USC president James Zumberge. Contact: Steve Gilbert, (609) 734-1915.

October 28 **Venture capital/Initial public offerings seminar** - San Franciso. Conducted by Accel Partners; sponsored by Adapso. With the voices of experience from Larry Michels (Santa Cruz Operation) and Kent Petzold (ViaSoft). Contact: Kelly Bailey, (703) 522-5055.

October 28-30 **AI conference** - Atlantic City, NJ. "Long Beach" in New Jersey. Call Jim Hay at Tower Conference Mgt., (312) 668-8100.

October 28-30 **Optical storage for small systems** - San Francisco (airport). Sponsored by Rothchild Consultants. Call Judith Hanson, (415) 681-3700.

November 2-5 **Electronic imaging '87** - Boston. Sponsored by Institute for Graphic Communication. Contact: Richard Murray, (617) 267-9425 (program) or Ed Martin, (617) 232-3976 (exhibits).

November 2-6 **COMDEX FALL** - Las Vegas. Needs no introduction... Contact: Jane Wemyss, Interface Group, (617) 449-0600.

November 13-15 **Hypertext '87** - Chapel Hill, NC. Sponsored by IEEE, ACM, and other worthy groups. Contact: Frank Halasz, co-author of NoteCards, (512) 338-3648, or John Smith, (919) 962-5021.

December 1-3 **Optical Information Systems** - New York City. Seminars and exhibits, focusing on CD ROM, sponsored by Meckler Publishing. Contact: Marilyn Reed, (203) 226-6967.

December 9-11
14-16 **Data base management systems for engineering and knowledge-based applications** - Santa Clara. Seminar course on object management systems led by Mohammad Ketabchi, with specific assessments of Vbase, GemStone and Iris. Contact Dr. Ketabchi at Santa Clara University, (408) 554-2731.

1988 (PLAN AHEAD!)

February 8-10 **IFIP conference on computers and law** - Santa Monica, CA. Issues that just won't go away: Copyright, contracts, taxation, computer crime, legislative actions. Sponsored by IFIP and Los Angeles County Bar Law and Technology section. Contact: Michael Krieger, (213) 208-2461.

February 17-19 **Software Development '88** - San Francisco. By programmers for programmers. With John Warnock, Gary Kildall, Dick Gabriel, and others. Sponsored by Computer Language and AI Expert Magazines. Contact: KoAnn Tingley, (415) 995-2426.

February 21-24 **ELEVENTH ANNUAL PERSONAL COMPUTER FORUM** - Naples, FL. We moved it in search of variety and better weather. Registration forms will be mailed to subscribers this fall. For further information, please call Sylvia Franklin, (212) 758-3434.

February 24-26 **Interactive instruction delivery** - Orlando, FL. Sponsored by Society for Applied Learning Technology. Call (703) 347-0055.

February 25-27 **Workshop on technology and cooperative work** - Tucson, AZ. Sponsored by Bell Communications Research and the University of Arizona. Contact: Robert Kraut, (201) 829-4513 or Jolene Galegher, (602) 621-7477.

March 8-10 **Connect '88** - New York City. Sponsored by Cahners, with Datamation and the Gartner Group. A trade show on connectivity and integration directed at corporate end-users. Contact: Richard Molden, (203) 964-0000.

March 14-18 **Artificial intelligence applications** - San Diego. Sponsored by IEEE. Contact: Richard Greene, (301) 468-3210 (exhibits) or IEEE, 371-0101 (program) or Paul Harmon (415) 861-1660.

March 16-23 **Hannover Fair CeBIT** - Hanover, West Germany. Contact: Donna Peterson Hyland, Hannover Fairs USA, (609) 987-1202.

March 23-25 **Conference on office information systems** - Palo Alto, CA. Sponsored by IEEE and ACM groups. With Terry Winograd and others. Contact: Robert Allen, (201) 829-5315.

March 23-31 **World Congress on Computing** - Chicago. Interface Group's response to NCC. Contact: Jane Wemyss, (617) 449-0600.

May 15-19 **Human factors in computing systems** - Washington, DC. Sponsored by ACM groups and the Human Factors Society. Contact: Sylvia Sheppard, (301) 369-2422, Scott Robertson, (201) 932-2911, or ACM (212) 869-7440.

May 31-June 3 **National Computer/Conference Exposition** - Los Angeles. NCC born again. Sponsored by AFIPS and managed by ISA Services. Conact: Philip Meade, (919) 549-8411.

August 22-26 **AAAI-88** - St. Paul, MN. The seventh annual. Sponsored by the American Association for Artificial Intelligence. Contact: Claudia Mazzetti, (415) 328-3123.

September 26-28 **Second conference on computer-supported cooperative work** - Portland, OR. Sponsored by ACM. Contact: Suzanne Sylvia, (617) 225-1860.

Please let us know of any other events we should include.

SUBSCRIPTION FORM

Please enter my subscription (at least 12 issues per year) to **Release 1.0** at the rate of $395 per year in the U.S. and Canada. Overseas subscriptions are $475, airmail postage included. Payment must be enclosed. Multiple-copy rates on request.

Name _____

Title _____

Company _____

Address _____

City_____ State _____ Zip _____

Telephone _____

87-8

Please fill in the information above and send with your check payable to:

EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

If you have any questions, please call us at (212) 758-3434.

Sylvia Franklin
Associate Publisher